CompSci 94
Classwork: Build a Logic Game
November 23, 2021

Prof. Susan Rodger

How to build this program

- Use the starter program but read through steps 1 and 2 to see what is in the starter program.
- Then follow the steps starting with step 3)

1) The Story

- This is a logic game. You will see three poles (actually ski’s) and there is an orange lantern at the bottom of each pole. Each lantern can move and either be “up” at the top of the pole or “down” at the bottom of the pole. The lanterns all start down. There is a secret code which is a position of all the lanterns such as “up” “down” “up”. You click on the lanterns to move them until you can get them in the position of the secret code. You are guessing the secret code!

1) Story continued…

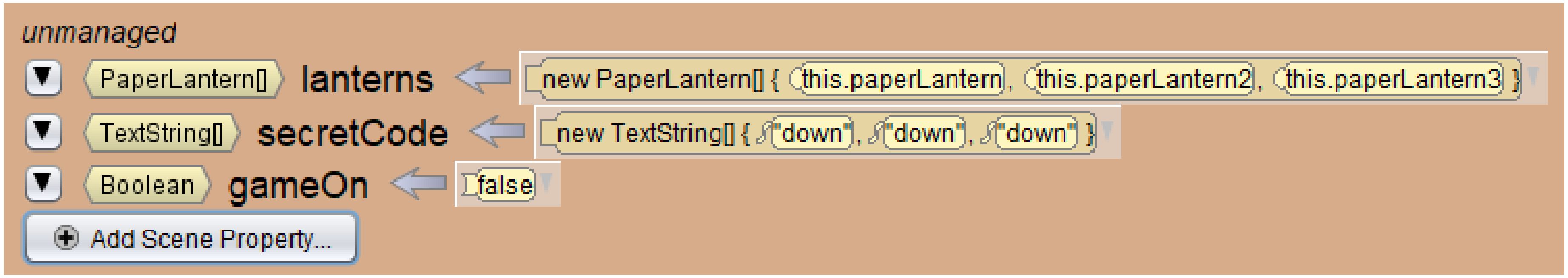
- You will click on the lanterns to move them to their other position (up or down).
- You get at most 5 clicks to get them in the position of the secret code.
- You will display how many clicks have happened.
- The game ends when you guess the code or you have five clicks and haven't guessed yet. Inform the user of which.

2) What is in the setup file?

- You get three skis: ski, ski2 and ski3 from left to right (width set to 0.13)
- You get three paperLanterns already put on the skis (we will call the skis poles). The paperLanterns from left to right are: paperLantern, paperLantern2 and paperLantern3 (all width set to 0.57)
- The lanterns are on the bottom of the pole, that is the starting position for the game.

2) Setup file continued

- In Scene properties you have:
  - an array named **lanterns** with the three paperLanterns
  - An array of type TextString named **secretCode** initially set to [“down”, “down”, “down”]
  - A Boolean variable named **gameOn** set to False

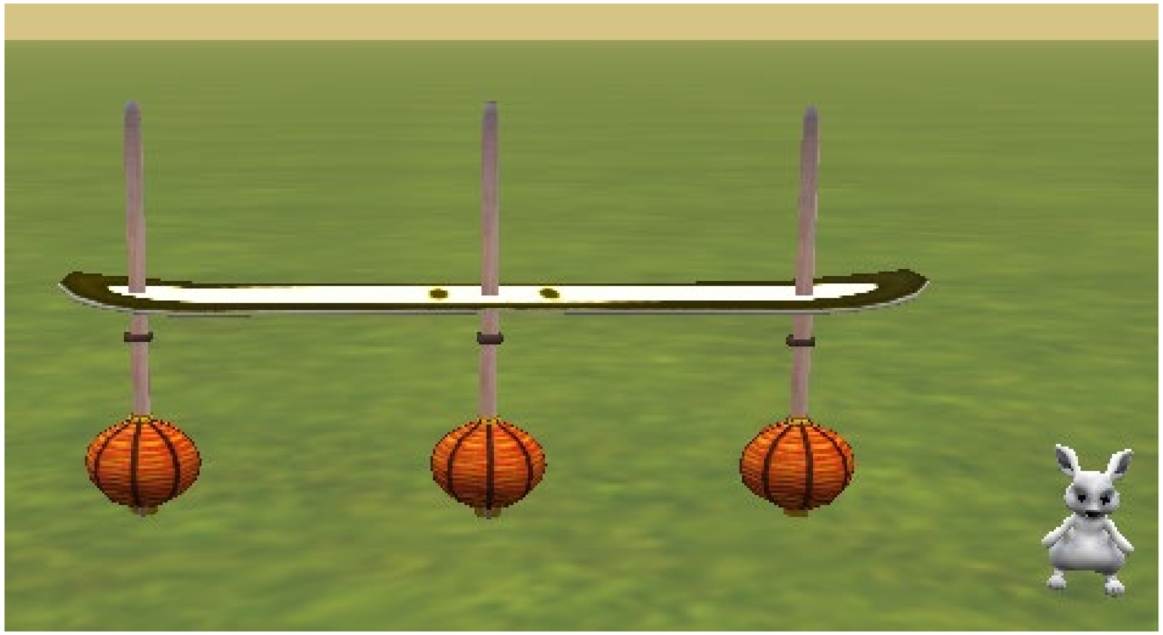


unmanaged

```
PaperLantern[] lanterns ⇐ new PaperLantern[] { this.paperLantern, this.paperLantern2, this.paperLantern3 }
TextString[] secretCode ⇐ new TextString[] { "down", "down", "down" }
Boolean gameOn ⇐ false
```

2) Setup file with **invisible** snowboard

- A snowboard (changed width to 0.8) has been added so you can tell whether or not the lantern is above or below it. It is invisible.



3) Write **Scene** procedure **generateSecretCode**

- This procedure has no parameters.
- You should loop through the secretCode array with an array index loop
  - For each word in the array, generate one of two random numbers and based on the number, set the new word in secretCode array to “up” or “down”

- For example, when this procedure ends, the secretCode array might be [“up”, “up”, “down”]

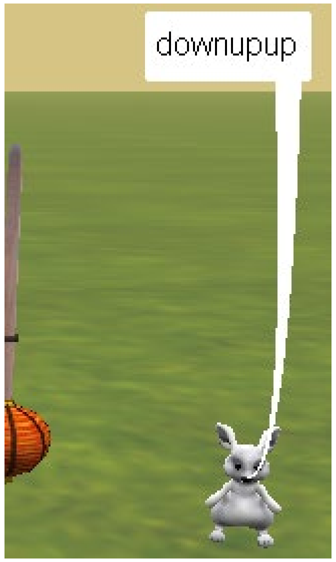
4) Write Scene Function getSecretCode

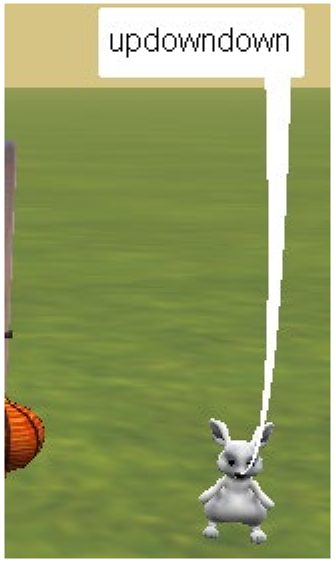
- This function has no parameters
- This function returns the secret code as a TextString.

- If the secretCode is [“up”, ”up”, “down”] then this function returns the string “upupdown”
- Loop through the array secretCode and build the string

5) Test your procedure and function

- In MyFirstMethod:
  - Call generateCode
  - Have the bunny say the secretCode (be sure to call getSecretCode)





- Run your program several times, you should get different results.

6) Make secret event to show code

- You don’t want to show the code before the user plays the game, but you do want to be able to see the code to see if your program is working.
- Delete the line where the bunny says the secret code.
- In the initializeEventListeners, create an event for **when you press “S”** the bunny says the secretcode. Test it out!

7) Create event to move lanterns

- Create an event so when you click on any lantern, that lantern moves up 1.5 units if it was at the bottom of the pole, otherwise it moves down 1.5 units.
- Move fast, duration 0.25!

- Test this out!

8) Write Scene Function checkForMatch

- This function returns a Boolean.
- It returns true if the lanterns are in the positions of the secretCode and otherwise returns false.

- Hint on how to do this on the next slide

8) Continued

- Loop over the lanterns array and for each lantern check its position. As you do this, build a string that matches the current configuration of the lanterns such as "updownup"

- Then compare the string you just built of the current positions of the lanterns to the string of the secretCode. If they match, return True! Otherwise return False!

9) Back in the event where you click on lanterns

- After moving a lantern, check to see if the lanterns match the secretcode. If they match then:
  - change the gameOn variable to false to turn off the game

10) Starting the game

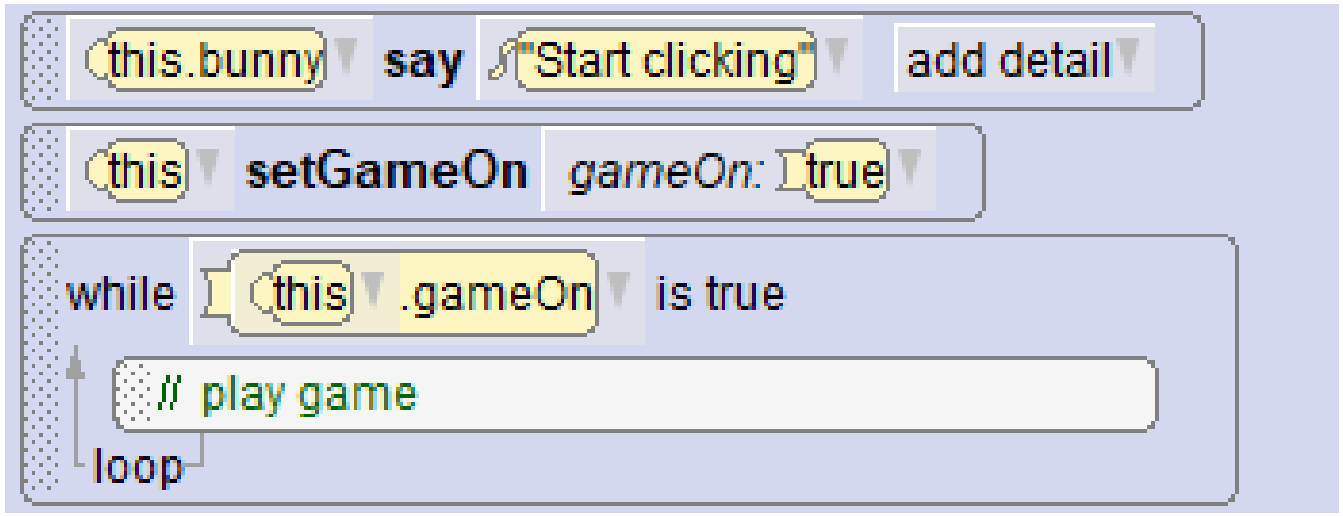
- Back in myFirstMethod:
  - You should have a call to generateSecretCode
  - However you don’t want to generate “downdowndown” or the game ends before it starts!
  - Put a while loop around the call to generateSecretCode and keep calling it until the secret code is NOT “downdowndown”

10) Continued in MyFirstMethod

- Have the bunny say “Welcome to the game”
- Ask if the user wants instructions and prompt the user to enter “yes” or “no”
- If they enter yes, then have the bunny say the instructions on how to play. Tell the user they need to click on the lanterns to move them until they match the secret code. Tell them they get up to five clicks

10) Continued in myFirstMethod…

- The bunny should say “start clicking”
- Have the gameOn variable set to true
- Add a while loop for while the gameOn is true



```
this.bunny say "Start clicking" add detail
this setGameOn gameOn: true
while this.gameOn is true
    // play game
loop
```

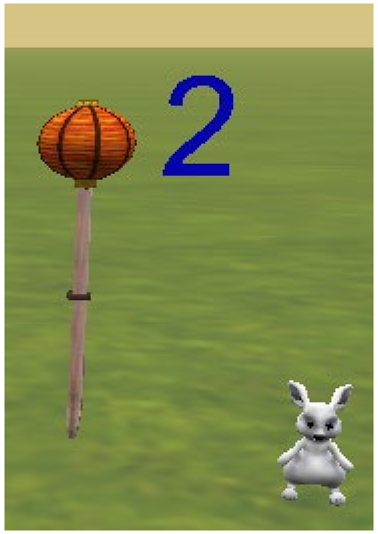
10) Continue MyFirstMethod

- After the while loop, check to see if the code matches the position of the lanterns (which function do you call?) and if they match, the bunny should say “You got it!”

- Play the game, press S to see the code and see if your game stops when you get the code.

11) Add a Scorer to show how many clicks the user has taken

- Add a scorer TextModel with a number property
- Include an updateScore procedure to add one to the score and redisplay it.



- After the user clicks on a lantern, update the score.

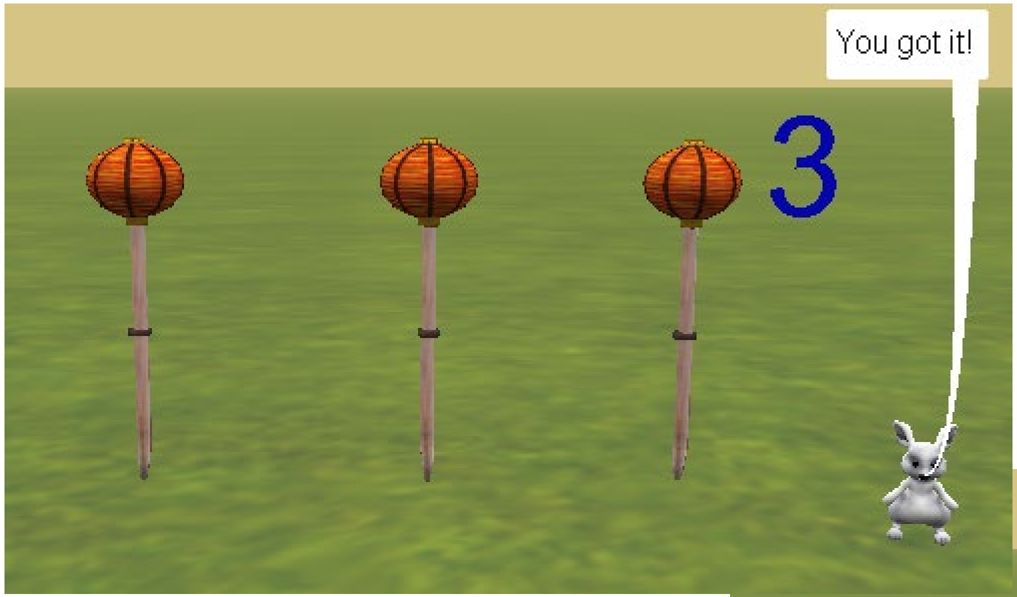
11) continued

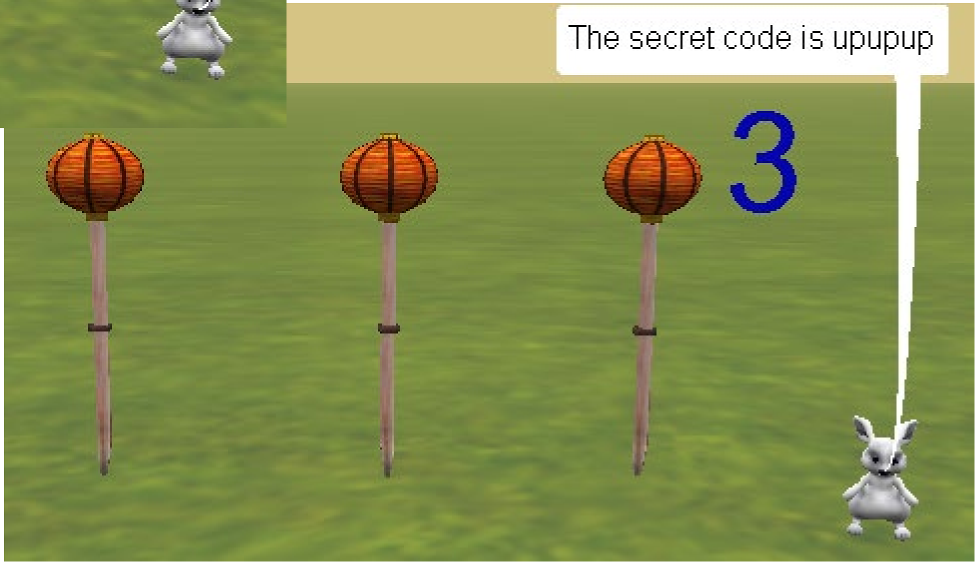
- Also in the mouse click event for lanterns, if there have been five clicks then set gameOn to false to turn off the game.

12) At end of myFirstMethod

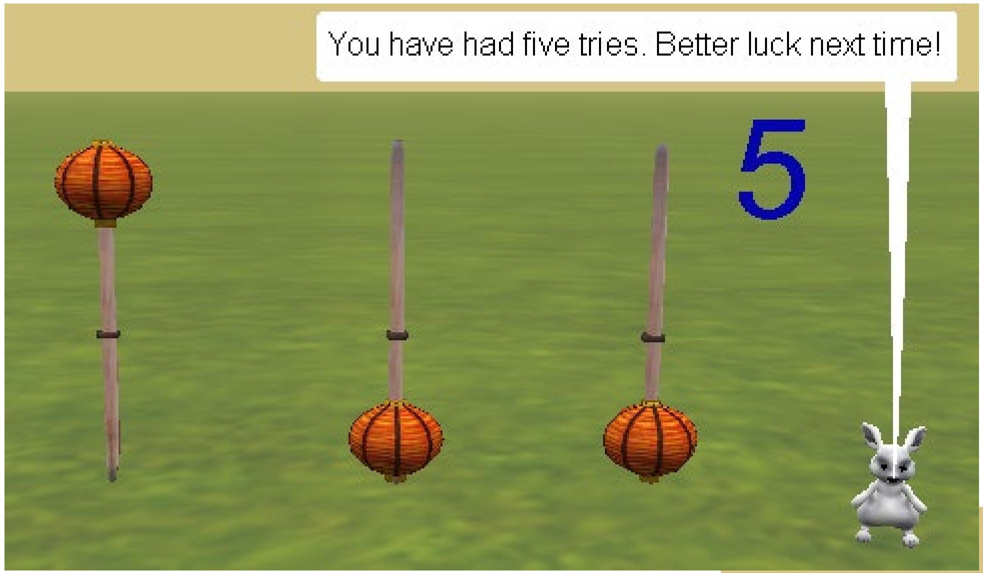
- When the game is over, if the code doesn’t match the bunny should say “You have had five tries, better luck next time.”
- Regardless of whether the code matched or not, at the very end the bunny should say what the code was.
- That’s it!

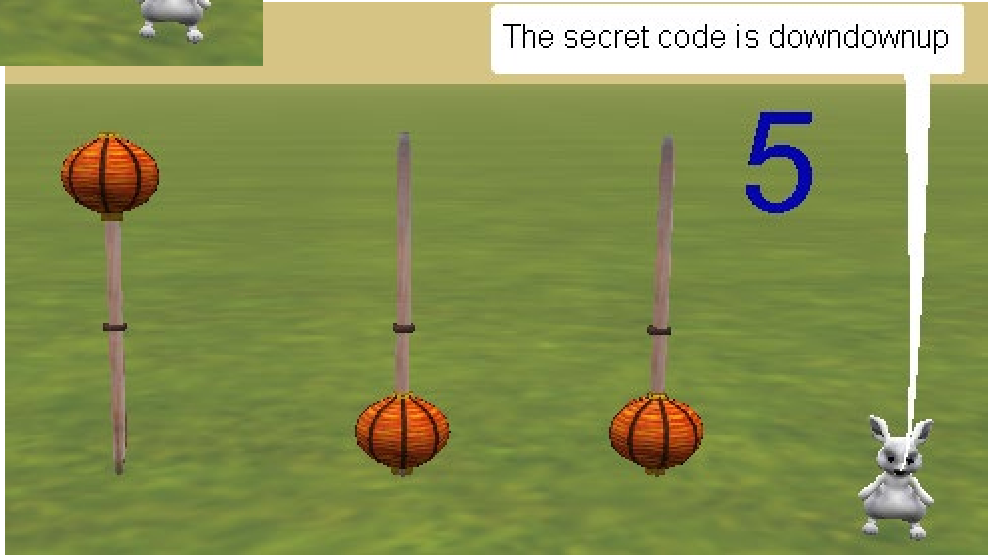
Game won with 3 clicks





Game losing 5 clicks


You have had five tries. Better luck next time!
5


The secret code is downdownup
5